Hidden Guardian

Design Document

Team 9
Client - Kelli Rout
Adviser - Swamy Ponpandi
Jennifer Frank - Team Lead
Keng-Yik Ho - Chief Engineer
Matthew Pedretti - Hardware Engineer
Thomas Kirby - Backend Programming
Jacob Stilwell - Computer Programmer
Team Email - sddec18-09@iastate.edu
Team Website - https://sddec18-09.sd.ece.iastate.edu/

Revised: 4/22/18
Version 2

Table of Contents

List of figures/tables/symbols/definitions

Not applicable

1 Introduction

1.1 Acknowledgement

Professor Swamy Ponpandi provided technical support and shall be acknowledged for assisting Team 9 in the development of Hidden Guardian. Similarly, Professor Zambreno and the SE, CPR E and EE departments provided financial support in the creation of this product.

1.2 Problem and Project Statement

Kid’s lives today involve online interactions more than ever. Their interactions with strangers invoke the risk of sharing unwanted personal information, password theft, receiving viruses, being cyberbullied and more. However, currently there is no practical parental monitoring system to keep track of these unwanted videogame interactions.

Our solution is Hidden Guardian. Hidden Guardian is a combination of a speaker/microphone for a gaming console. It will connect to the XBox Controller as well as the Hidden Guardian's mobile app and speaker. Through the mobile app the parent user will be able to input keywords, such as meet, phone number, address and will be notified if they are used in chats. The app will also provide live text streams of the child’s conversations and specific text paragraphs of chats with those keywords. Hidden Guardian will only be able to be enabled or disabled by the parent.

Overall, our goal is to provide a system so that parents can have peace of mind that their kids videogame interactions are safe.

1.3 Operational Environment

The typical operating environment of Hidden Guardian is a household setting. Given that houses are typically well climate controlled the hardware component won’t be subject to any extreme conditions. It will however be handled frequently and should be at least somewhat robust in order to withstand moderate impacts.

1.4 Intended Users and Uses -

The intended user for this project will be for 2 targets, parents and their children. The children will be able to use the bluetooth speaker for their gaming uses or listening to music. The speaker will provide immersive gaming experience by providing high quality audio. The device will also have a microphone to enable children to communicate with other players online to improve gameplay.

Parents will be using it as a device to monitor their children's activities. By using the microphone on board, the device will record all of the child's conversations while the device is turned on. Parents will be able to access logs of recordings using the app and search for keywords in the recording such as phone numbers, and addresses to help improve the safety of their children. Additionally if the child is using a headset the parent will be able to use the speaker to listen to their child's live conversations.

1.5 Assumptions and Limitations

Assumptions:

- At the end of development, this product will not be commercially ready to consume
- The maximum amount of hours/week of data we will store is 15.
- We don't need to comply with licensing rights, since we are handling the data outside of the console.
- Hidden Guardian is used in a household

Limitations:

- For speech to text recognition we will only support one language (English).
- Our device won't be able to separate each voice and associate it with different users
- We won't be able to access text chat communications
- The device will require relatively large amounts of power for a battery powered device
- Hidden Guardian's Speaker can only be used in Parent Mode up to 60 feet away

1.6 Expected End Product and Deliverables

- Smartphone App

  The smartphone app will allow users to review and search through the data collected by the console app. It will have a user friendly interface and provide a variety of tools for filtering through the stored data. These tools include things like being able to search for keywords or users and having all relevant conversations pop up.

- Hardware Speaker

  The wireless speaker will have a rechargeable battery and a bluetooth receiver. The speaker will play audio data received via its bluetooth receiver, and will user a usb cable to recharge its battery.

- Hardware Controller Extension

  The controller extension will house a microphone and a set of controls for both the microphone and speaker. The controls will be buttons affecting microphone and speaker volume, as well as muting the microphone. The extension will attach to the bottom of an Xbox one controller using the 3.5mm jack. The extension will also contain a 3.5mm jack itself to serve as a passthrough for any user wanting to use their own headset. The extension will also house a bluetooth and wifi transmitter for connecting to the speaker and database respectively.

- Database

  The database will receive audio data and convert it into text. Both the audio data and text transcription will be stored in database tables.

The initial stages of each of the mobile app and Database will be delivered by May and the final versions of those components as well as the hardware will be delivered by the beginning of December.

2. Specifications and Analysis

2.1 Proposed Design

Our proposed design consists of four main parts. A controller attachment will contain a microphone and series of controls for both the microphone and wireless speaker. Additionally the attachment will have a Raspberry Pi with bluetooth and wifi capabilities to enable transmitting chat audio to the speaker, and both chat and microphone audio to a database after the audio is converted from analog to digital using the sound card adapter. The attachment will connect to the bottom of the console's controller via a 3.5mm audio jack, and will have its own 3.5mm audio jack if the user wants to connect their own headset. The wireless speaker will have a bluetooth receiver and adjustable volume. The speaker will also come with a code that will allow the user to associate the speaker with their personal account. The database will convert the audio into text and store both the audio file and the text transcription. An android application will provide a UI for interfacing with and retrieving data from the database. The user will have a username and password to connect to their data in the database. The app will have search tools which will allow the user to filter through the gamer's chat data via text transcripts and will notify the user/guardian based on specific keywords they have set up. Because speech to text software can make translation mistakes, especially with variable volume levels and microphone qualities, we are going to provide flagging for the confidence level of each word or phrase of a translation. Combining the ability to see translations of low confidence with the ability to listen to the relevant audio recording should allow the user to more accurately assess the conversation data.

To help visualize this design refer to Figures 1 and 2.

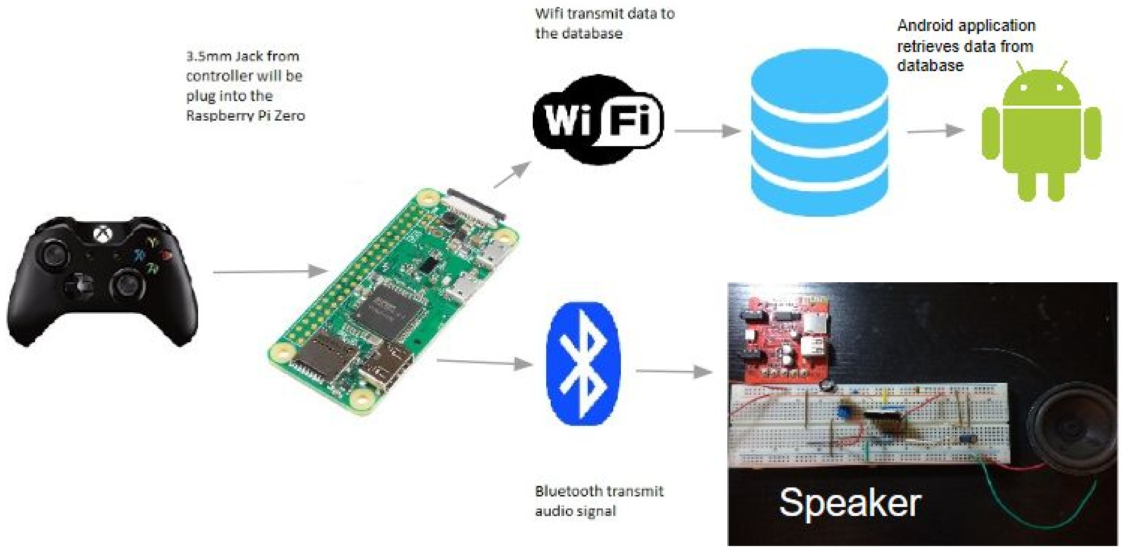
3.5mm Jack from controller will be plug into the Raspberry Pi Zero
Wifi transmit data to the database
Android application retrieves data from database
Bluetooth transmit audio signal
Speaker

Figure 1 - Conceptual Sketch

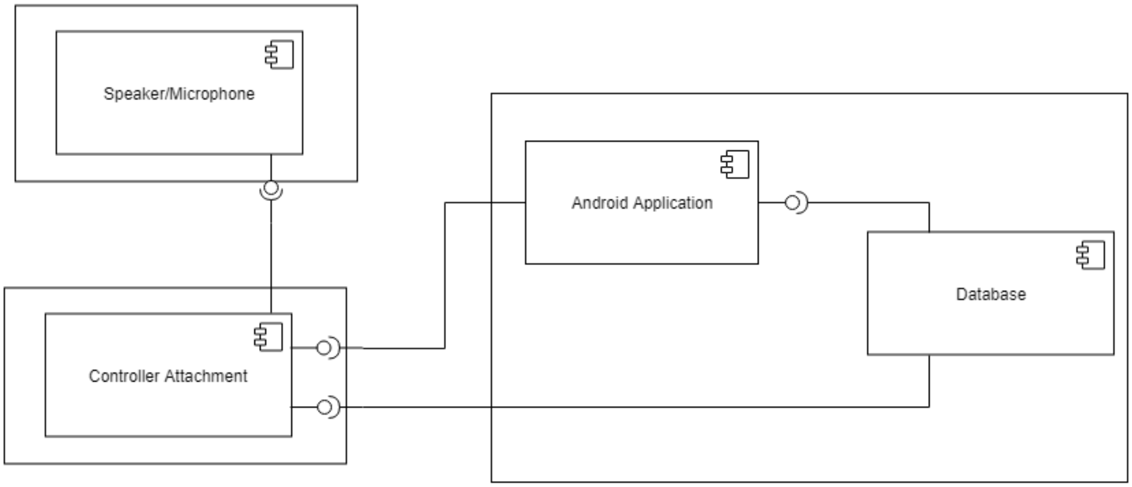

Figure 2 - Block Diagram

Functional Requirements: The speaker device will have the function to output audio data coming from the headjack and double as a wireless speaker that outputs the live data including the microphone output. The conversations will be stored on a database which can only be accessed by an account tied to the device. The database will be able to be searched for user-specified keywords using an app. The speaker must be wireless, rechargeable and bluetooth capable. The controller attachment must have a microphone that records data.

Non Functional Requirements: Our focus on nonfunctional requirements include efficiency/speed, interoperability, usability, security and scalability. Our main user is parents who could be of ages 25-50 and will have varying technological understanding. A requirement that is of utmost priority is to make the mobile application easy to understand and use. We also will focus on speed and efficiency with our applications, so that data is being sent at appropriate time intervals, there is not a slow buffer time or cached incorrect data. Our product needs to be interoperable so that the purchase of a speaker contains a code to access the console application which can smoothly connect to a database which connects to a mobile app all synced together. We will also need to have interoperability between the speaker and the XBox One Controller Attachment via wifi. Security is an important requirement so that we are making sure the text data is only viewable by the parent and their username/password data is protected. We also need to make sure that our product is scalable so for future potential production it can be extended to iOS apps, web applications, PS4's and other gaming systems. Since we cannot test this, we will have to focus on having excellent documentations, commenting and making sure our code is modular and easy to understand and pass on.

Current Project Status: Currently, we have a Background application that has functional speech to text capabilities. Though we will not be using this it has prepared us for using a different speech to text tool next semester. Our mobile application has login front end and due to recent issues an almost implemented backend. We have a speaker prototype that can connect to bluetooth and amplify sound. We have also tested the speaker prototype and know its uninterrupted bluetooth range is 30 feet. Our android application team members have made it past the learning curve for using Android Studios and we have a better understanding of our database capabilities. There are now diagrams that clearly outlined plans for our future hardware implementation and team members now have a clear vision and understanding of how to implement our project with the recent changes.

Standards: For standards relating to privacy and security of data we will meet general standards of encryption of data being sent to and from the database and applications. However, before this product is able to go into production it will need a firewall system and further more intensive privacy and security implementations that will be handled by our client.

One thing that we've considered with ethical standards is whether it is ethical to distribute a chat record of audio conversations between one's child and potential other children or strangers. Though this is of concern, it falls under the same category as other gaming streaming services. Those who sign XBox's privacy license are willingly consenting to a potential distribution. I also think it is worth considering that this product's main consumers will be parents of children whose main intent behind purchasing is to keep their kids safe.

2.2 Design Analysis

So far we have done research into the feasibility of our initial design. In our research we encountered two major issues with our initial design. These issues were that the audio data from a week of online chatting would be excessively large for a phone application, and that the xbox does not natively support bluetooth communication. To get around the data size issue we modified our plan to convert the audio data into text using the database rather than the phone, significantly reducing the amount of data that needs to be downloaded onto the phone.

Our plan to get around the issue with the wireless speaker is to implement a second hardware module that will connect to the controller that will communicate with the speaker via bluetooth. This second hardware module will connect to the 3.5mm audio jack on the controller that is used for headsets. The controller module has the added benefit of being a convenient location to implement easy to reach buttons as well as a more convenient place for a microphone. A downside of having to use the audio jack is that commands can't be sent to the console to affect the volume or mute settings in software. This could be solved by using the expansion port on the controller but that presents its own challenges as that port is proprietary and it would be extremely difficult to develop a prototype for.

3 Testing and Implementation

3.1 Interface Specifications

The controller attachment will plug in to the 3.5 mm audio jack of a standard Xbox one controller. The audio signal from the audio jack will be sent through usb port of the Raspberry Pi Zero W. The Raspberry Pi will send the signal via bluetooth to the wireless speaker, and via WiFi to the database. The microphone on the controller attachment will be sent both to the Raspberry Pi via usb and to the controller's audio jack. The controller attachment will also have a 3.5 mm audio jack passthrough. The android application will access the database to display information to the user. Refer to Figure 1 as it shows the connections of these interfaces visually.

3.2 Hardware and software

Hardware

Bluetooth Module

- Receive audio for the speaker.

Raspberry Pi Zero

- To send the audio or mic signal to the bluetooth speaker and connect to the database through wi-fi

Raspberry Pi sound card

- Converts analog audio into a digital format

TDA7266 Dual Bridge Amplifier

- A class AB amplifier to amplify the signal received from the Bluetooth module and send to the speaker.

Software

SQL database

- Store audio files
- Convert audio files into text data using Javoix
- Store converted text data
- Store datetimes
- Store encrypted usernames and passwords
- Store keywords

Android app

- User login with account tied to an individual speaker
- Access text and audio data from the database
- Dynamic table of keywords for searching through the database

Due to our change in approach decided by the team, adviser and client, we have decided to not have a console application. This was replaced with Hardware.

3.3 Functional Testing

Examples include unit, integration, system, acceptance testing

- Testing whether speaker can receive a bluetooth signal and amplify it based on the volume setting
- Testing the distance at which the bluetooth speaker can stand without losing a signal
  - We will test this by sending audio data with a phone (and later the XBox and mic via the controller attachment) and see how far we can take the speaker without it breaking a signal uninterrupted
  - We will do the same with the speaker but interrupted with ceilings and walls and report those statistics
- Test whether the controller extension can pair with the speaker via bluetooth
- Test the speakers multifunctional ability
  - When the speaker is off, the extension is still receiving data and sending it to the mobile application
  - When the speaker is in “child mode” it is outputting only the data from the 3.5 mm jack
  - When the speaker in “parent mode” it is outputting the data from the 3.5 mm jack and the microphone data
  - When the speaker is in “parent mode” the microphone is off
- Test transmitting an audio signal to the speaker through the controller extension by sending a distinct audio file and confirming the speaker outputs the same audio file (tested manually)
- Test receiving microphone data to the controller extension
  - We will have multiple voices at multiple volumes speak while using the controller and confirming that the controller extension receives it.
  - We will test the retrieval of the data via the speaker and the database/mobile applications audio and text data.
- Test dumping all data on app into database

    - Use corner cases by writing tests that dumb 100,000 usernames and making sure the database maintains its performance
- Unit tests on database server to make sure querying works
    - We can test this by querying SQL to insert a table or gather data from a table and create unit tests that the right results appear.
- Testing specific set of keyword system using unit tests
    - Test the ability to select a suggested keyword, add a keyword, delete a keyword
    - Test the ability to pick up all keywords by parsing through the text data
    - Test the reaction time that it takes to realize there is a keyword from the time that it was spoken and then notify the parental user (non functional requirement)
    - Test that the notification shows up for the user (manually)

3.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

- Will test the most intuitive location layout for the microphone and speaker controls.
    - We can do this with software that will create multiple possibilities and find the best one for placing our layout
- Test that our controller extension fits comfortably in the space between controller handles
- Will test our login feature to guarantee security
    - Unit tests for password requirements such as length, and amount of uppercase, numeric and special characters
    - Unit tests for someone creating an account with an existing username
    - Unit tests for when a user wants to link to a speaker's ID that already exists
    - Unit tests for when a parent user tries to access data for a speaker that they are not connected to
- Will test compatibility between database and other components
    - We can test this by querying SQL to insert a table or gather data from a table and create unit tests that the right results appear.
- Will test the performance of the speaker and its reaction time to receiving live data
    - We will do this by sending a distinct audio file through the XBox headset and manually listening for a noticeable lag. We will be doing this manually because if the lag isn't audible by the human ear then it meets our standards. We will also be doing this from 5 feet intervals ranging from 0 to 30 feet away.

3.5 Modeling and Simulating

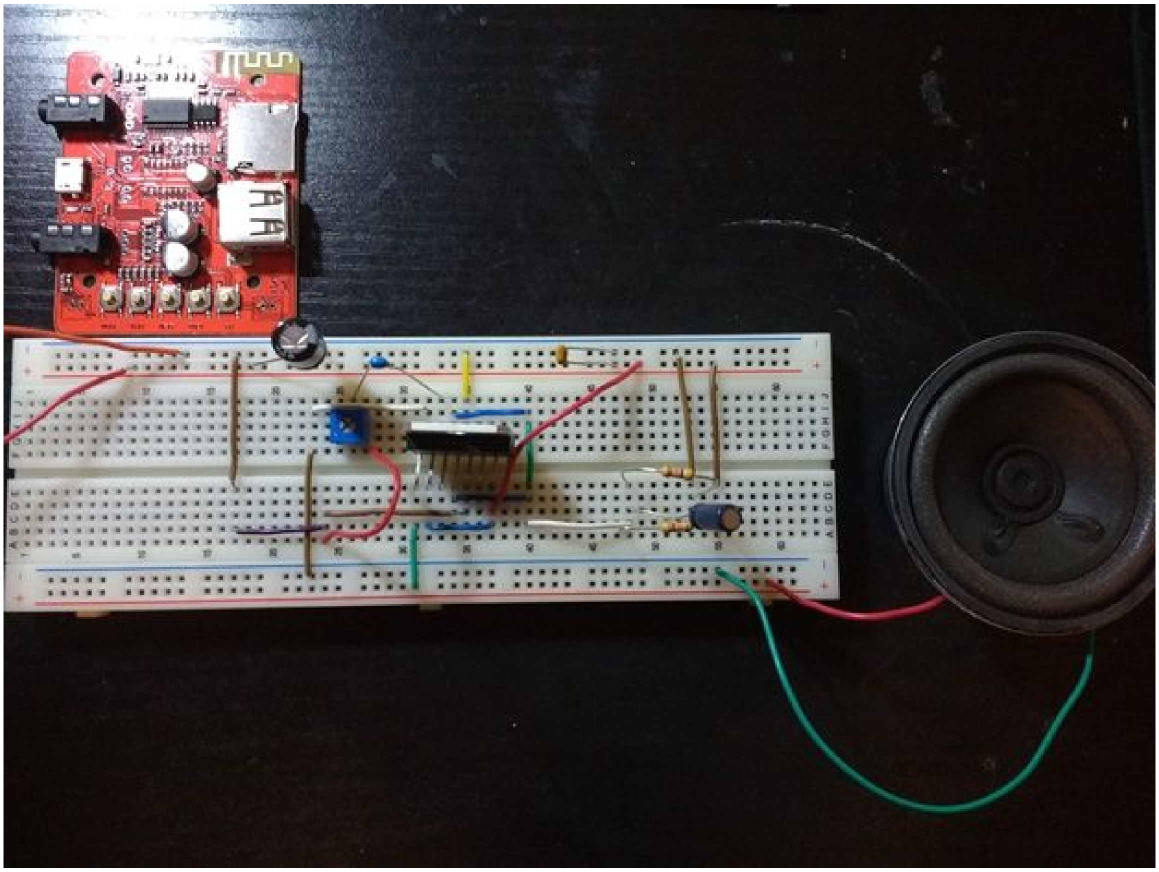

Figure 3 - A working prototype of the Bluetooth speaker

We plan to model our bluetooth wireless speaker and simulate our mobile application functionality.

3.6 Process

Due to the phase we are in of our timeline we have just started the testing process. We have tested whether the speaker can receive a bluetooth signal and amplify it (not based on volume settings quite yet though). We've done this by pairing our phone with the speaker's bluetooth and sending audio data. We've also tested the distance at which our bluetooth speaker can stand and it is 30 feet uninterrupted. As described in our Test Plan we will later test the distance our speaker can stand without breaking a bluetooth connection while interrupted (such as with walls or ceilings/floors as this may be applicable to a household. Though we will not be later implementing our console application, we have tested the speech to text functionality by speaking into the system at varying volumes and distances to see how it affects the confidence ratings. We've found that while speaking casually at a distance similar to what the microphone will be from the user's mouth, the confidence rating is labeled "moderate" of the three ratings, low, moderate, high. We have not yet been able to test our mobile application but we will at the beginning of next semester. For GUI testing we will need to have manual tests that we can consistently check off because for Java Application automated testing applications tend to be too expensive (based on prior research).

Once are project begins development we will have figures, images, examples and a much more thorough description of our testing process for every deliverable we will provide.

3.7 Results

Challenges and Issues

Challenge 1: Inability to use XBox's proprietary bluetooth transmitter

Challenge 2: Inability to get the game and party chat audio data from the console

Challenge 3: Inability to get the text chat data from the console

Challenge 4: Inability to use bluetooth to connect the microcontroller to the database (we will need to use Wifi which is less user friendly for set-up).

Current Issue 1: Have to resend the command on the Raspberry Pi every time when it powers down

Current Issue 2: Power consumption issue with the controller attachment

Current Issue 3: Learning Jaivox and implementing Speech to Text within the database

Results

We failed to realize that the XBox has no bluetooth transmitter so we realized we will need to construct hardware ourselves to account for that. We also realized late in the semester that the console would not give us access to the game chat data directly and that we would have to find an alternative solution. We successfully came up with an alternative solution to implement a man-in-the-middle type wiretap to access the audio data. We have successfully developed the first prototype of our wireless speaker as well as our android app. We have learned how to connect our various components with our database and plan on refining the structure of the database to meet all of our requirements. We've also learned how to implement speech to text programs and have a mockup running on console that is simulating what the database will be doing for us in the final version of our project.

4 Closing Material

4.1 Conclusion

Our goal of this project is to have a platform to monitor children when they are playing games so they would not disclose any information that would jeopardize their safety. We plan to have a speaker and a controller attachment with a microphone and microcontroller that could potentially replace the gamers headset, and a mobile application. Between the communication of these components the parental user will be able to listen for keywords, listen to chats, and effectively monitor their child's video game play on a user-friendly mobile app.

4.2 References

Speaker Amplifier

http://www.st.com/resource/en/datasheet/tda7266.pdf

Raspberry Pi Zero W

https://www.raspberrypi.org/products/raspberry-pi-zero-w/

Android Application

https://www.androidauthority.com/android-app-development-complete-beginners-658469/v
http://www.vogella.com/tutorials/MySQLJava/article.html
Financial Resources
https://www.gamestop.com/browse/consoles/xbox-one?nav=28-xu0,13ffff2412-1e0
https://www.xbox.com/en-US/developers
Speaker Research
https://www.parts-express.com/cat/midrange-midbass-drivers-full-range-speakers/16

https://www.digikey.com/
Battery Life Research
https://www.neogaf.com/threads/how-does-the-xbox-one-controllers-battery-life-compare-to-the-dualshock-4s.759677/
IEEE Ethical Code
https://www.ieee.org/about/corporate/governance/p7-8.html
Microsoft Windows UWP Forums

https://docs.microsoft.com/en-us/windows/uwp/

4.3 Appendices

Not applicable.